Generischer Architekturansatz für Telemedizin Portale und verteilte Krankenakten

Wolfgang Wiedermann[1,3], Athanassios Tsakpinis[3], Christian Wolff[2]

[1)] Klinikum der Universität Regensburg

[2)] Universität Regensburg

[3)] Fachhochschule Regensburg

1 Abstract

Dieser Aufsatz beschäftigt sich mit Softwarearchitekturen und Kriterien für deren längerfristige Verwendbarkeit als Basisansatz für die Entwicklung von verteilten Krankenakten und Telemedizin-Portalen. Im Vorfeld der Entwicklung eines verteilten Krankenaktensystems mit Portalfrontend am Klinikum der Universität Regensburg werden hierfür verschiedene Architekturansätze untersucht. Dabei soll ein Architekturprinzip gefunden werden, das unabhängig von Technologien und Standards die Entwicklung flexibler und gut integrierbarer verteilter Krankenaktensysteme ermöglicht. Nichtfunktionale Anforderungen der gesuchten Lösung sind Veränderbarkeit, Anpassbarkeit, Zuverlässigkeit, Erweiterbarkeit und Fehlerrobustheit. Das hier vorgestellte Konzept wurde während der Vorstudien zum Aufbau des Portalsystems entwickelt; bisherige Erfahrungen aus dem Betrieb des Portals stützen die dort getroffenen Annahmen. Als zentraler Anwendungsfall für die beiden genannten Systemtypen wird die gemeinsame Nutzung medizinischer Dokumente innerhalb einer stark heterogenen Systemlandschaft vorausgesetzt.

2 Rahmenbedingungen

Im nationalen und internationalen Umfeld gibt es eine vergleichsweise große Anzahl unterschiedlicher Standards wie z.B. IHE XDS (Integrating the Healthcare Enterprise – Cross Enterprise Document Sharing, vgl. [1]), HL7-CDA (Health Level Seven – Clinical Document Architecture) und eFA (elektronische Fallakte, vgl. [2]), die sich dem Thema verteilter Krankenakten widmen. Aufgrund ihrer hohen Komplexität sind sie nur in wenigen in Deutschland verfügbaren Produkten implementiert und decken jeweils nur einen Teilaspekt des Problems ab – dies allerdings i. d. R. umfassend. Etabliertere Standards wie HL7 v2 oder DICOM (Digital Imaging and Communications in Medicine) bieten im Hinblick auf Sicherheit und Verfügbarkeit nicht die notwendigen Eigenschaften wie sie für hoch verteilte Szenarien benötigt werden. Auch die Betrachtung verschiedener Lösungsansätze (wie z.B. [3] und [4]) zeigt, dass eine hinreichend allgemeine und gleichzeitig praktikable Lösung des Problems der verteilten Nutzung von Krankenakten noch nicht existiert. Die zunehmende organisatorische Vernetzung von Gesundheitsdienstleistern erfordert jedoch eine adäquate Informationstechnologische Vernetzung, d.h. die effiziente Weitergabe von behandlungsrelevanten Daten. Dies wird durch die große Heterogenität der Systemlandschaft sowohl im Haus, als auch bei den beteiligten Kommunikationspartnern (Kliniken, Facharztpraxen etc.) aber auch durch die noch relativ große Unklarheit bezüglich im realen Betrieb relevanten Benutzeranforderungen [5] erheblich erschwert.

Heterogene, aber technisch noch nicht voll umgesetzte Standards einerseits, und der Zwang zu stärkerer Integration der IT-Systeme andererseits ergeben ein Spannungsfeld, das die Notwendigkeit der Entwicklung einer von Standards, der Systemlandschaft aber auch von Details der Benutzeranforderungen unabhängigen Systemarchitektur verdeutlicht.

3 Architekturansatz

Das in 1 beschriebene Spannungsfeld erfordert eine Systemarchitektur, die im Kern den Zugriff auf sämtliche Dokumentenarten vereinheitlicht, so dass mit vergleichsweise geringem Entwicklungsaufwand zusätzliche Schnittstellen, basierend auf beliebigen Standards für, das System konsumierende Clients zur Verfügung gestellt werden können. Außerdem muss die Architektur auch für den Zugriff auf die Quellsysteme (KIS, RIS usw.) die Möglichkeit der Kommunikation mittels unterschiedlicher Standards und sogar proprietärer Protokolle vorsehen. Ein drittes Kriterium ist, dass es, beispielsweise für den Fall des Dokumentenuploads, auch prinzipiell möglich sein muss, dass das System auch selbst Dokumente verwaltet und speichert.

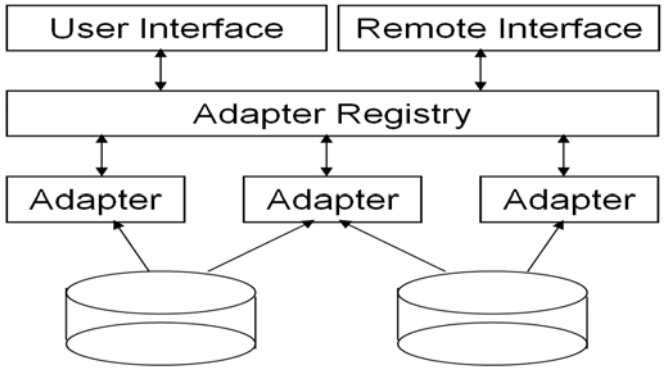

Abbildung 3: Architektur - Grobdarstellung

Dies alles lässt sich durch die in Abbildung 1 dargestellte Architektur realisieren. Die Unifikation des Dokumentenzugriffs erfolgt dabei in der *„Adapter Registry Komponente“* welche die Quellsystem-Adaptoren verwaltet und deren Funktionalität über eine innerhalb der Anwendung standardisierte Schnittstelle den Schnittstellenkomponenten (hier *„User Interface“* und *„Remote Interface“*) zur Verfügung stellt. Die *„Adapter Komponenten“* realisieren den technischen Zugriff auf die Quellsysteme und adaptieren dabei das Verhalten der *„Adapter Registry Schnittstelle“*. Aufbauend auf der Schnittstelle der *„Adapter Registry“* können sowohl direkt eine Benutzeroberfläche angeboten werden als auch verschiedene Remote-Schnittstellen basierend auf beliebigen Standards der Remote-Kommunikation wie beispielsweise SOAP, RMI oder DICOM zur Integration mit Drittsystemen angeboten werden.

Um den Aufbau der im Rahmen dieses Textes beschriebenen Softwarearchitektur detailliert zu beschreiben um sie auch in Einzelkomponenten wieder verwendbar zu machen wird diese nun anhand ihrer, den Softwareengineering-Prinzipien „Patternorientierte Softwarearchitektur“, „Softwarekomponenten“ und „Serviceorientierte Architekturen“ entnommenen Artefakte betrachtet.

4 Patternorientierte Betrachtung

Durch die konsequente Verwendung von Architektur- und Entwurfsmustern bei der Spezifikation der vorliegenden Softwarearchitektur kann diese als „Patternorientierte Softwarearchitektur“ [6] betrachtet und beschrieben werden. Die dabei verwendeten Architekturmuster sind zum einen die „*Mikrokernel Architektur“* [7] und zum anderen die „*Schichtenarchitektur“* [7], die sich hier aufgrund der unterschiedlichen Granularität der Komponenten in Kombination zur Erreichung einer verständlichen und gut erweiterbaren Systemstruktur anbieten.

Mikrokernel:

Grundprinzip der vorliegenden Softwarearchitektur ist eine, dem Anwendungsfall „Verteilte Krankenakte“ angepasste Mikrokernel-Architektur.

Der Nutzen einer Mikrokernel-Architektur wird sehr gut durch die folgende Aussage beschrieben: „The Microkernel architectural pattern applies to software systems that must be able to adapt to changing system requirements. It separates a minimal functional core from extended functionality and customer-specific parts. The microkernel also serves as a socket for pluggin in these extensions and coordinating their collaboration”[7].

In dieser Architektur ist der Mikrokernel Kernbestandteil der in Abbildung 1 mit dem Namen “Adapter Registry” bezeichneten Komponente und verwaltet die Zusammenarbeit der verschiedenen Adapterimplementierungen sowohl untereinander als auch mit diversen Hilfsdiensten, wie z.B. der Logging- oder der Authentifizierungs-Komponente. Durch die Anwendung dieses Musters wird erreicht, dass das System ohne Veränderung bereits Qualitätsgesicherter existierender Komponenten um neue Komponenten erweitert werden kann. In der am Klinikum der Universität Regensburg verwendeten Implementierung ist das Hinzufügen von Komponenten via Konfiguration auch zur Laufzeit möglich.

Schichtenarchitektur:

Die Verwendung des Schichtenarchitektur-Musters dient der Aufteilung der Komponenten anhand ihrer Aufgaben und ermöglicht so eine Anschauliche Strukturierung der durch die Anwendung des Mikrokernel-Architekturmusters entstanden Ansammlung von Komponenten (siehe Abbildung 2).

- Client Schicht: Applikationen die i.d.R. genau eine der Komponenten der Provider Schicht konsumieren, und so beispielsweise einem Benutzer den Zugriff auf die Dokumente der am System angebundenen Quellsysteme ermöglicht.

- Provider Schicht: Die Komponenten der „Provider Schicht“ dienen der passiven Weitergabe der Daten aus dem System an sie konsumierende Drittsysteme.

- Kernkomponente: Bündelt die Dienste der Adapterschicht zu einem semantisch äquivalenten Dienst und erweitert diesen um Authentifizierung, Usermapping usw.

- Adapter Schicht: Stellt durch jeweils einzelne Adapterimplementierungen die Funktionalität der Quellsysteme in Form einer für alle Adapterimplementierungen verbindlichen Schnittstelle zur Verfügung. Dabei greifen die Adapterimplementierungen aktiv auf ein oder mehrere Quellsysteme zu. Der dabei verwendete Kommunikationsstandard orientiert sich jeweils an den Fähigkeiten des Quellsystems. Das heißt, dass ein PACS z.B. via DICOM, ein SAP System entweder über Webservices oder SAP-RFC oder ein MEDOS System über HL7 angebunden werden kann. Dabei wird in der Beziehung zwischen Adapterimplementierung und Kernkomponente das Adapter-Pattern aus [8] angewandt.

- Quellsysteme: Unter Quellsysteme werden die Systeme verstanden, die die Dokumente real verwalten und sich i.d.R. bereits im jeweiligen Krankenhaus in Betrieb befinden.

Verwendung von Entwurfsmustern:

Um die beschriebenen Eigenschaften bezüglich Flexibilität und Erweiterbarkeit zu erreichen wurden im Rahmen der Implementierung Entwurfsmuster wie das *„Proxy Pattern“* [8] und das *„Component Configurator Pattern“* [9] verwendet. Sie dienen im Wesentlichen der Entkopplung der verwendeten Komponenten und ermöglichen so die bereits erwähnte, per Deployment und Konfiguration mögliche Austauschbarkeit zur Laufzeit.

5 Komponentenorientierte Betrachtung

Wie bereits mehrfach erwähnt basieren sowohl die Flexibilität als auch die Erweiterbarkeit der hier vorgestellten Architektur auf der Austauschbarkeit von Komponenten. Außerdem ist dem Komponentengedanken auch eine Verbesserung der Übersichtlichkeit des Gesamtsystems zu verdanken.

In Abbildung 2 wird die Systemarchitektur durch ein Komponentendiagramm dargestellt, in dem auch die Anwendung der in 2.1 beschriebenen Architektur-Muster „Mikrokernel“ und „Schichtenarchitektur“ optisch erkennbar wird. Außerdem wurde eine strikt aufgabenorientierte Zerlegung des Gesamtproblems in Komponenten durchgesetzt, welche zu aussagekräftigen Komponentennamen und zu im Innenverhältnis übersichtlichen Komponenten führte. Im Außenverhältnis können die dargestellten Komponenten, auch aufgrund ihrer eindeutigen Aufgabenzuordnung als *„Black-Box“* behandelt werden. Das heißt, dass es ausreichend ist, wenn den Entwicklern anderer Systemkomponenten die Schnittstelle der Komponente bekannt ist.

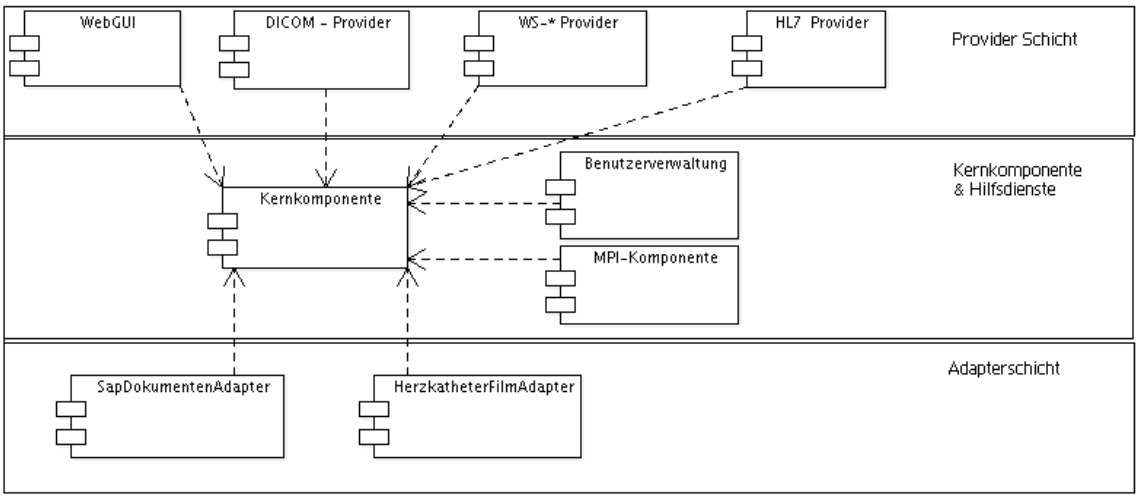

Abbildung 4: Architekturansatz als Komponentendiagramm

Die Verwendung von komponentenorientierter Middleware-Technologie wie z.B. JEE 5 oder .NET macht es möglich, dass die beschriebenen Komponenten auch unabhängig voneinander auf verschiedene Server verteilt lauffähig sind und mittels entfernten Aufrufen miteinander kommunizieren können.

Da jede Adapter-Implementierung eine eigenständige Komponente ist lässt sich ein System dieser Architektur durch hinzufügen und entfernen entsprechender Adapterkomponenten in beliebige Systemlandschaften integrieren.

5.1 Serviceorientierte Betrachtung

Die in 2.2 beschriebenen Komponenten stellen im Außenverhältnis Schnittstellen zur Verfügung, die grundsätzlich die formal an Dienste im Sinne einer SOA (Serviceorientierte Architektur) gestellten Anforderungen [10] erfüllen. Andererseits ist aus Sicht des Unternehmens Klinikum auch das gesamte System „verteilte Krankenakte“ mit seinen Operationen zur Suche und Übertragung von medizinischen Dokumenten ein Service. Gemeinhin wird im Bereich SOA zwischen fachlichen- und technischen Services unterschieden, wobei fachliche Services einer fachlichen Operation also beispielsweise einem Prozessschritt entsprechen. Technische Services hingegen behandeln Querschnittsaufgaben, die nicht der Benutzer von Systemen, sondern die Systeme selbst benötigen und gemeinsam Nutzen. Ein Beispiel für einen technischen Dienst ist in unserem Fall die Schnittstelle zur Komponente „Benutzerverwaltung“ aus Abbildung 2.

Das Gesamtsystem stellt einen fachlichen Dienst zum Retrieval und zur Übertragung von Patientendaten und medizinischen Dokumenten aus einer Menge von angebundenen Subsystemen über mindestens folgende Operationen dar.

- Patienten finden und auflisten
- Patientendaten laden
- Dokumente finden und auflisten
- Metadaten eines Dokuments laden
- Dokument laden

Gleiches gilt für die einzelnen Adapterkomponenten, wobei sich der Wirkungsbereich der Operationen dabei auf genau ein angebundenes Subsystem beschränkt.

6 Einsatzszenarios

Im folgenden Abschnitt wird anhand zweier Einsatzszenarios beschrieben wie das in 2 beschriebene Architekturmodell praktisch als Basis für ein Krankenakten-Portalsystem und für eine verteilte Krankenakte verwendet werden kann.

6.1 Anwendung im Rahmen eines Portalsystems

Unter Krankenakten-Portalsystem wird hier ein System verstanden, das einem fest definierten Benutzerkreis mittels einer webbasierten Benutzeroberfläche auch außerhalb eines Krankenhauses sicheren Zugriff auf Dokumente der dortigen elektronischen Krankenakte erlaubt. Diese Dokumente können beispielsweise verteilt auf verschiedene Subsysteme vorliegen.

Hierbei handelt es sich um den Anwendungsfall in dem die vorliegende Architektur bereits im Rahmen unseres Pilotbetriebs erprobt wurde. Dabei werden Befunddokumente aus dem SAP ISH*Med System, PDF Dokumente aus dem Dokumenten-Management-System und Herzkatheterfilme und -standbilder aus dem PACS über eine einheitliche webbasierte Benutzeroberfläche bereitgestellt.

6.2 Anwendung im Rahmen einer verteilten Krankenakte

Eine verteilte Krankenakte liegt in diesem Zusammenhang vor, wenn die hier beschriebene Softwarearchitektur verwendet wird, um Dokumente aus einer Menge von Quellsystemen eines oder mehrerer Gesundheitsdienstleister sowohl schreibend als auch lesend über eine gemeinsame Schnittstelle nutzbar sind. Dies kann beispielsweise auch bedeuten, dass der Zugriff in den verschiedenen beteiligten Institutionen über unterschiedliche Client-Software geschieht.
Hier eignet sich die vorliegende Architektur beispielsweise dazu, in verschiedenen Systemen, welche unterschiedliche Kommunikationsstandards unterstützen Dokumente zu Verwalten und diese über die Serviceschnittstelle als HL7 CDA Dokumente zu übermitteln. Hierfür kann in den Implementierungen der Adapter-Komponenten die Übersetzung von und nach dem ursprünglichen Ausgangsformat des jeweiligen Quellsystems erfolgen.

7 Ausblick

Im Rahmen des Projekts wurden diese und andere Architekturvarianten zwischen April 2006 und April 2007 durch verschiedene prototypische Implementierungen auf ihre Fähigkeiten im Bezug auf Erweiterbarkeit, Integrierbarkeit, Entwicklungsaufwand und Flexibilität getestet. Die oben dargestellte Softwarearchitektur wies dabei die aus unserer Sicht besten Eigenschaften auf, was dazu führte, dass sie die Basis für unser seit Januar 2008 mit ausgewählten Partnern im Betrieb befindliches System wurde. Auch im Pilotbetrieb erwies sie sich als stabil und gut erweiterbar. Vor allem die medienbruchfreie Integrierbarkeit des Systems in die Prozesse zur Erstellung von Befunddokumenten führte in der Praxis dazu, dass nahezu alle fertiggestellten Befunddokumente einschließlich ihrer Begleitdokumente für ihre Adressaten via Portal zugängig sind. Für den langfristigen Erfolg derartiger Systeme ist aber auch eine sehr enge Integration in die Systeme der externen Partner notwendig, was nur durch eine deutliche Verbesserung der Konnektivität im Bereich der Praxissysteme möglich ist.

8 Literatur

[1] Cross-Enterprise Document Sharing-b (XDS.b); IHE IT Infrastructure Technical Framework Supplement 2007-2008; Draft for Trial Implementation August 14, 2007 http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_Supplement_XDS-2.pdf; Geladen 11.02.2008

[1*] Kurzbeschreibung zu IHE XDS; Geladen 11.02.2008; http://wiki.ihe.net/index.php?title=Cross_Enterprise_Document_Sharing

[2] Caumanns, J; Electronic Case Records for Intersectoral Cooperation - Core Concepts and Architecture Overview; Fraunhofer ISST Berlin; Berlin 26.09.2006 http://www.fallakte.de/download/eFA_SolutionSummary_v1.0.pdf

[3] van der Haak, M; Knaup-Gregori, P; Skonetzki, S.; Architekturkonzepte für einrichtungsübergreifende elektronische Patientenakten; in Proceedings zur Telemed 2006; Berlin 2006

[4] Bayer, A; Prokosch, U.; Web Plattform der Universitätsklinik Erlangen für die Umsetzung telemedizinischer Anforderungen und Dokumentation integrierter Versorgungssysteme; in Proceedings zur Telemed 2007; Berlin 2007

[5] Bergmann J; Zur Architektur transinstitutioneller elektronischer Patientenakten zur Unterstützung vernetzter Versorgungsinfrastrukturen, Bremen 2006

[6] Stal, M.; Software-Muster; In Reussner, R.; Hasselbring, W. (eds.) Handbuch der Software-Architektur. Heidelberg: dpunkt 2006, 331-355.

[7] Buschmann F, et al; A System of Patterns. Pattern oriented Softwarearchitecture, Volume 1; Chichester: Wiley and Sons 1996

[8] Gamma E; Entwurfsmuster: Elemente wieder verwendbarer objektorientierter Software; Aus dem Amerikan. Von Dirk Riehle – Bonn: Addison-Wesley, 1996

[9] Schmidt D, Stal M, et al; Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, Volume 2; Chchester: Wiley and Sons 2000

[10] Josuttis N; SOA in Practice – The Art of Distributed System Design, First Edition, O´Reilly, August 2007